



(12) 发明专利申请公开说明书

[21] 申请号 90104994.8

[51] Int.Cl⁵

G09G 5/22

[43] 公开日 1991年3月13日

[22] 申请日 90.7.26

[30] 优先权

[32] 89.7.28 [33] US [31] 387,569

[71] 申请人 德克萨斯仪器股份有限公司

地址 美国德克萨斯州

[72] 发明人 尼古拉斯 K·英格-西蒙斯

伊恩 C·罗伯逊

[74] 专利代理机构 上海专利事务所

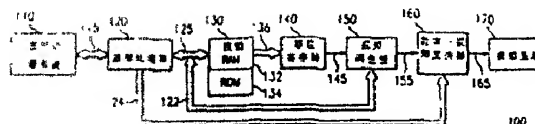
代理人 颜承根

说明书页数: 17 附图页数: 8

[54] 发明名称 图形显示对分式串行寄存器系统

[57] 摘要

一种与图形显示器的像素大小或每行上的像素数目无关的图形数据显示电路和方法。存储器输出的对分式串行寄存器与计数器一起用于维持对寄存器当前工作的输出级进行计数。当寄存器的第一半已完成把它的数据传送给显示部分时,它被清零且用存储器的下一行的第一半重新对它装入。其第二半以上述同样方式工作,并用上述存储器行的第二半对其重新装入。这种交替操作允许在不影响数据传送性能的情况下在一行的中途对寄存器进行刷新。



(BJ) 第1456号

权 利 要 求 书

1.一种图形处理系统包含:

图形显示介质 (a graphics presentation media);

一行一行地传送数据到所述介质的串行寄存器; 所述寄存器包含
轮换进行工作的第一和第二部分, 每部分在另一部分传送数据期间都可重新装载;

对传送给所述介质的信息的存储地址进行追踪的监控寄存器; 和

由所述监控寄存器带动的控制电路, 用于控制把数据传送给所述寄存器的所述第一和第二部分。

2.如权利要求 1 所述的图形处理系统, 其特征在于所述监控寄存器装入介质行的新行中第一个数据位置的存储地址, 且其后在所述数据传送给所述介质行期间, 所述监控寄存器增量计数到一个预先规定的计数, 由此交替控制所述寄存器的所述第一和第二部分。

3.如权利要求 2 所述的图形处理系统, 其特征在于所述要传送的数据包含在存储器的行和列中, 并且数据从所述存储器的存储单元传送给所述串行寄存器, 可采取一次送一整行的方式, 此时所述串行寄存器的所述第一和第二部分分别包含来自给定行的第一半和第二半数据, 也可采用半行方式, 所述寄存器的所述第一部分或所述第二部分分别包含来自相同行或不同行的第一半或第二半的数据。

4.如权利要求 2 所述的图形处理系统, 其特征在于进一步包含:

规定的移位寄存器掩蔽码; 和

分接点移位寄存器包含代表所述第一个数据位置的数位, 所述分接点数位可通过将所述移位寄存器的掩蔽码与表示包含所述存储地址的行和列地址数据的数据位图进行匹配来产生。

5.如权利要求 4 所述的图形处理系统, 其特征在于进一步包含:

根据从所述串行寄存器传送来的数据对所述监控寄存器进行控制的增量电路。

6.如权利要求3 所述的图形处理系统，其特征在于进一步包含，根据所述半行方式控制所述数据移入所述串行寄存器的行地址寄存器。

7.一种把数据从显示存储器(memory for presentation) 移到总线的方法，所述数据可在行和列的基础上寻址。通过把寄存器对分成两部分，每一部分都可以从这部分传送数据，同时，允许向另一部分装入数据，所述方法包含步骤：

在所述存储器的任一行中的任一列位置接受所述存储器中的一个存储单元地址以便从所述行把数据提供给所述总线；

从所述行把所述数据传送给所述对分式寄存器；

控制数据从所述对分式寄存器一位一位地输出传送到所述总线上，所述控制部分地是通过对所述对分式寄存器中的数据的数据的所述列位置进行计数进行的；

在所述接受的存储单元地址的所述列位置的基础上，计算出在所述对分式寄存器中的起始位的位置

从所述计算出的(起始) 位位置开始并根据对所述计数的列位置的控制、交替地把所述对分式寄存器的这一部分或另一部分的数据传送到所述的总线。

8.如权利要求7 所述方法，其特征在于进一步包含步骤：

当属于所述接受行的数据正在从所述对分式寄存器的第二部分送出时，从所述存储器的下一个半行把数据传送到所述对分式寄存器的第一部分

9.如权利要求8 所述方法，其特征在于所述总线上出现的所述数据将按照一个像素接一个像素的方式立即送到视频显示器，且所述的

最后提到的向所述对分式寄存器进行的传送是在所述视频显示器的回扫区间由把数据传送到所述视频显示器期间完成的。

10. 如权利要求7所述方法, 其特征在于所述计算包含步骤:

设计一个掩蔽码;

产生代表所述起始位位置的分接点, 通过把所述的掩蔽码对包含所述行和列地址数据在内的要显示的数据的位图相匹配来产生所述的分接点, 且所述控制步骤包含:

从与所述分接点对应的所述对分式寄存器中的位置开始对所述对分式寄存器的位置进行计数。

11. 如权利要求10所述方法, 其特征在于进一步包含步骤:

把所述分接点的二进制值装入计数器; 和

对所述计数器增量计数, 加上对所述对分式寄存器所述数位的检索(indexing of said bits), 共同控制所述传送步骤。

12. 图形处理系统的工作方法, 包含:

图形显示介质; 和

用于一行接一行地传送数据到所述介质的串行寄存器; 所述寄存器包含轮换操作的第一和第二部分, 每一部分都可在另一部分向外传送数据期间重新装载; 所述系统包含对送到所述介质的信息的存储单元地址追踪的步骤; 和

按照所述追踪的存储单元地址控制数据传送到所述寄存器的所述第一和第二部分。

13. 如权利要求12所述方法, 其特征在于进一步包含步骤:

对介质的一个新行的第一个数据位置的存储单元进行装载; 和

在数据通过所述传送到所述介质期间, 增量到预定的计数值, 由此对所述寄存器的所述第一和第二部分进行交替控制。

14. 如权利要求13所述方法, 其特征在于所述要传送的数据包含

在存储器中的行和列中，数据以一行接一行的整行方式从所述存储器的存储单元传送到所述串行寄存器，使所述串行寄存器的第一部分和第二部分分别包含所给定行的第一半和第二半，或根据半行方式，使所述寄存器的第一部分或是第二部分分别包含来自相同行或不同行的第一半或者第二半的数据。

图形显示对分式串行寄存器系统

本发明涉及图形显示系统，尤其涉及这样一种使用对分式串行寄存器(asplit - serial register)以控制数据传送到图形显示器的系统。

在视频图形系统中，屏幕上的像素数和必须从显示存储器中读出的点数是很重要的。使用显示存储器数量的多少对任何系统的经济利用是一个关键因素。在典型的系统中，显示存储器中存有一个必须在短时间间隔内读送到屏幕的数据行。上述过程由与存储器相连接的一个串行寄存器来完成、该寄存器把整行数据从存储器中移出并把该行传送到屏幕上。

人们总是希望尽可能把存储器存满并使用整个存储空间。也希望所设计的存储器系统可适应不同的图形需求和实际限制的变化。对于目前的存储器系统，通常使用每行长度具有512位的视频随机存取存储器(VRAM)。这些存储器有几块板，通常每个存储器有四块板，并且对于每个存储器的读周期，这种存储器都安排得填满总线。于是，对于32位总线，可使用8块VRAM，每个VRAM有4块板。

按照这种结构，每块存储器的每块板的串行寄存器为512位长，因此，对于每个存储器到串行寄存器的读周期，将传送16384位到显示器。假定每个像素包含8位，则这些16384位代表了2048个显示像素的数据。假设每根扫描仅需要1280个像素，则对于存储器的每行而言，将有768个像素不能显示。这些存储单元也难用作其它用途，因此造成浪费。

一种考虑过的利用整个存储器行的安排，是把这剩余的768个像

素用于下一行。而根据这种设计，在信息送往屏幕期间，必须对寄存器重新装载。除非使用特定的高速硬件，否则这种在一行的中间对寄存器重新装载会使观察者看得见而难以接受。

对这种安排的一个改进是使用一对并行的移位寄存器，使数据在其中交替读出。这种安排也遇到刚刚讨论过的困难，即在于移位寄存器的大小也取决于预期的屏幕的大小。

于是，在技术上需要这样一种串行寄存器图形重新装载系统，它允许在图形显示屏幕的一行上连续地装填像素，而与屏幕行的长度无关，因而节省了存储器和寄存器的空间。

在图形处理系统中使用了一种对分式串行寄存器 (a split - serial register) 以控制屏幕传送信息，使在每行输出期间信息能连续地从该移位寄存器移出，从而不间断地向屏幕提供数据流。这种系统允许在行的中间重新装载而不会使观看者察觉。

串行寄存器的两半部分在水平消隐 (行间回扫) 屏幕刷新周期期间装入数据，与此同时把行的起始点的列地址提供给地址寄存器。当消隐期间结束并时钟开始时，数据从相应于列地址的位开始串行地从数据寄存器的一半输送到屏幕上。然后串行寄存器的地址进行逻辑计数与寄存器的数据移位一起进行)，从起始地址变到全部1，再翻转到全部为0。在零点上，数据移位自动地从另一半串行寄存器开始。这允许当前不用的半个寄存器用于接收与屏幕信息的下一部分相对应的数据。

先前未用的一半寄存器现在开始工作，重复执行同样的过程其地址寄存器的计数也是先到全部为1，再翻转到全部为0。当地址再次达到零时，已重新装载了的该行后续的新信息便从另一半寄存器开始移出它的数据。此时，新的数据又提供给串行寄存器的另一半。这种交替操作连续进行到该行图形行结束为止，把连续的信息流送到屏幕，即

使在行中间进行更新也不会使用户察觉到。

潜在的问题在于，如果新行的起始点超过串行寄存器的中点时，那么闲置的一半寄存器、即串行寄存器的第一半包含着与先前行末尾有关而与当前行的下面部分无关的信息。于是该第一半寄存器必须更新以保证在第二半寄存器卸载之后，使第一半包含下一组跟随的数据点。这一问题的解决是在正常的水平消隐屏幕刷新之后，为对分式串行寄存器增加一个刷新周期，并立即产生下半行的地址。该刷新周期与用于行中间重新装载的屏幕刷新周期类型相同，且仅装载每个串行寄存器的一半。

使用对分式串行寄存器有效地产生到视频屏幕的连续的数据流有着技术优点。当寄存器的一部分向屏幕送数据时，寄存器的另一部分从存储器接收数据。一个地址控制寄存器用来对与串行寄存器和屏幕有关的数据的存储位置保持追踪。

通过下面结合附图进行的描述，很容易理解本发明的上述目的和其它目的，其中：

图1 表明根据本发明的原理构成的具有图形处理能力的计算机的方框图；

图2 表明本发明图形处理电路的较佳实施例的方框图；

图3 表示与对分式串行寄存器一起工作的视频存储器的扩大的等效视图以表示本发明的一个实施例；

图4 表示用于说明目的的图形显示器；

图5 表示用于说明目的的存储阵列；

图6、7和8 表示在不同时间串行寄存器内部各位的情况；

图9和10（表示在第一张纸上）显示两种可能的对不同尺寸的存储器的行地址和列地址的安排；

图11、12和13表示按照不同的实际寻址结构对串行寄存器的分接

点 (tap point) 进行控制掩蔽码位,

图14表示用于计算对串行寄存器的控制的算法和流程图, 和

图15—21表示与本发明的说明实例相一致的控制寄存器中的位分布。

在详细讨论本发明之前, 参看图1 和图2 简略地回顾一下与主机系统一起工作的图形处理器的基本工作是有帮助的。至于它的详细讨论能从申请日为1989年4月27日的 No. 346,388 专利申请 (代理人道特 (D O C K e t) N O T 1—9484 B) 中找到, 并且已转让给该申请的受让人。因此该申请可作为背景资料。这里一起作为参考的还有: 德克萨斯仪器公司的用户指南 T M S34010 和 T M S34020, 以及说明书 T M S34020/ T M S34082 和 T M S44C251。所有这些资料目前都可以从德克萨斯仪器公司获得。

为了方便和容易理解这里指出的本发明的概念, 不打算表述每步操作及每个数据的流动, 因为在任一系统中使用的本发明的实际实施例, 在很大程度上取决于该系统实际操作。在这样的系统中本发明的概念将被具体化。

图1 表示按照本发明的原理所构成的图形计算机系统100 的方框图。图形计算机系统100 包含主机处理系统110、图形处理器120、存储器130、移位寄存器140、视频调色板150、数字到视频变换器160 和视频显示170。

主机处理系统110 给图形计算机系统100 提供主要的计算能力。主机处理系统110 最好包含至少一个微处理器、只读存储器、随机存取存储器和各种外围装置以构成一完整的计算机系统。主机处理系统110 最好也包含一些输入装置如键盘或鼠标器, 和一些长期存储装置如盘驱动器等。主机处理系统110 的详细构成基本上是属传统的已知技术, 因此本申请将不详细叙述它的构成。就本发明而言, 主机处理

系统110 的主要特征在于它确定要向用户显示的视频显示器的内容。

根据本发明，图形处理器120 提供主要的数据处理，以产生要向用户显示的特定的视频显示。图形处理器120 经主机总线115 与主机处理系统110 进行双向耦连。按照本发明，图形处理器120 作为独立于主机处理系统110 的数据处理器而工作；然而，要求图形处理器120 能经主机总线115 响应来自主机处理系统110 的请求。图形处理器120 进一步通过视频存储器总线125、122 与存储器130 和视频调色板150 进行通信。图形处理器120 经视频存储器总线125 控制存储器在视频RAM132 中的数据。另外，也可以通过存储在视频RAM132 中或只读存储器134 中的程序，对图形处理器120 进行控制。只读存储器134 还可以包含各种类型的图像数据、如常用的图标符号，以及一种或多种字体的字母数字字符等。此外，图形处理器120 也控制存储在视频调色板150 中的数据。最后，图形处理器120 经视频控制总线124 控制数字到视频变换器160。通过经视频控制总线124 对数字到视频变换器160 进行的控制，图形处理器120 可以控制向用户显示的每帧视频图像的行数和行长。

视频存储器130 包含经视频存储器总线125 双向耦连到图形处理器120 的视频RAM132 和只读存储器134。如前所述，视频RAM132 包含控制向用户显示的视频图像的位映射图形数据。这种视频数据可通过视频存储器总线125 由图形处理器120 进行处理。另外，与当前显示屏相对应的视频数据从视频RAM132 经视频输出总线136 输出。视频输出总线136 的数据与要向用户显示的像元相对应。在较佳实施例中，视频RAM132 由德州仪器的受让人德克萨斯仪器公司可供应的多个动态随机存取集成电路TM S44251 256 K×4 组成。TM S44251 集成电路含有双端口，使显示的刷新和修改不会互相干扰。

移位寄存器140 接收视频 RAM132 的视频数据并把这些数据组成显示器的位数据流(bit stream)。按照视频随机存取存储器132 的典型安排, 这种存储器是由几个分开的随机存取存储器集成电路组成的一个存储体, 每个这种集成电路的典型输出宽度只有一位或四位。因此有必要把多个这种电路的数据进行组合, 以获得足够高的输出速率来规定向用户显示的图像。移位寄存器140 以并行方式从视频输出总线136 装入数据。然后以串行方式从线145 输出。于是移位寄存器140 组成显示用的位数据流, 由这数据流对光栅扫描视频显示中的各个像点提供速率足够高的视频数据。

视频调色板150 经总线145 接收来自移位寄存器140 的高速视频数据。视频调色板150 也经视频存储器总线122 接收图形处理器120 的数据。视频调色板150 把从总线145 上接收到的数据变换成总线155 上的视频输出电平。这种变换是用经由视频存储器总线122 传送的在图形处理器120 中定义的一个查找表来完成的。视频调色板150 的输出可包含每个像素的色调和饱和度, 或可包含每个像素的红、绿和兰基色电平。从储存在视频存储器132 中的编码到经总线155 输出的数字数据电平的变换表、经视频存储器总线122 受图形处理器120 的控制。

数字变视频的变换器160 经总线155 接收视频调色板150 的数字视频信息。数字变视频的变换器160, 经视频控制总线124 、受图形处理器120 的控制。数字变视频的变换器160 把视频调色板150 的数字输出变换为所需要的模拟电平, 经视频输出165 用于视频显示器170 。数字变视频的变换器160, 经视频控制器总线124, 由图形处理器120 控制, 例如用来规定每帧的行数和每水平行的像元数。图形处理器120 中的数据、通过数字变视频的变换器160 控制产生同步和消隐信号及回扫信号。这部分视频信号不是由存储在视频存储器132 中的数据规

定的，而是产生一组必要的控制信号，以规定所需的视频输出。

最后，视频显示器170 经视频输出线165 、接收数字变视频的变换器160 的视频输出。视频显示器170 通过图形计算机系统100 的操作者产生所规定的可视视频图像。应该注意：视频调色板150 、数字变视频的变换器160 和视频显示170 可按照两种主要的视频技术工作。第一种技术是对各个像素规定其彩色色调和饱和度的视频数据。另一种技术是规定每个独立像素的红、绿、兰的基色电平。根据所采用的是哪一种技术的设计选择，使视频调色板150 、数字变视频的变换器160 和视频显示器170 的构造适合于这种技术。然而，本发明的图形处理器120 的工作原理是不变的，不论视频技术选择哪种特定的设计。

图2 进一步详细表示了图形处理器120 。图形处理器120 包含中央处理单元200 、专用的图形硬件210 、寄存器文件220 、指令高速缓冲存储器230 、主机接口240 、存储器接口250 、输入/输出寄存器260 和视频显示控制器270 。

图形处理器120 的核心是中央处理单元200 。中央处理单元200 具有在通用的中央处理单元中常规具有的包括各种算术和逻辑运算操作在内的通用数据处理能力。另外，中央处理单元200 具有许多专用的图形控制指令，这些指令可单独应用或结合专用图形硬件210 应用。

图形处理器120 包含与图形处理器120 大多数部件包括中央处理单元200 相连接的主总线205 。中央处理单元200 经双向寄存器总线202 双向耦连到一组包括许多数据寄存器的寄存器文件。寄存器文件220 用作中央处理单元200 的即时存取数据的贮存所。象下面将要详细叙述的那样，寄存器文件220 除了包含中央处理单元可能用列的通用寄存器组外，还包含许多用于贮存图形指令隐含操作数的数据寄存器。

中央处理单元200 经指令高速缓冲存储器总线204 与指令高速缓

冲存储器230 相连。指令高速缓冲存储器230 进一步耦连到总线205, 且可经过视频存储器总线122 和存储器接口250 从视频存储器132(图1)装入指令字。指令高速存储器230 的用途是加速执行中央处理单元200 的某些功能。凡经常用于由中央处理单元200 执行的程序的特定部分中的重复功能可贮存在指令高速存储器230 中。经指令高速存储器总线204 从指令高速存储器230 中存取比从视频存储器130 中存取快得多。于是, 由中央处理单元200 执行的程序, 通过预先把重复或常用的指令序列存入指令高速存储器230 中可以提高处理速度。这些指令由于它们可快速取出从而它们可快速执行。指令高速存储器230 并非总是包含相同的指令步骤, 但总是装入在中央处理单元200 所执行的程序的特定部分中经常会用到的一组特定指令。

主机接口240 经主机接口总线206 耦连到中央处理单元200。主机接口240 经主机系统总线115 进一步与主机处理系统110(图1)相连接。主机接口240 用来在主机处理系统110 和图形处理器120 之间进行通信。主机接口240 在主机处理系统110 和图形处理器120 之间控制数据定时传送。在这方面, 主机接口240 能使主机处理系统110 中断图形处理器120, 或反之亦然, 能使图形处理器120 中断主机处理系统110。另外, 主机接口240 被耦连到主总线205, 而能使主机处理系统110 直接控制存储在存储器130 中的数据。通常, 主机接口240 可把主机处理系统110 的图形请求送到图形处理器120, 从而使主机系统能指定由视频显示器170 产生的显示类型, 并且使图形处理器120 完成所需的图形功能。

中央处理单元200 经图形硬件总线208 耦连到专用的图形硬件210。专用的图形硬件210 进一步连接到主总线205。专用的图形硬件210 与中央处理单元200 一起工作以执行专门的图形处理操作。中央处理单元200, 除了它的通用数据处理功能外, 还控制专用图形硬件

210 的应用以便执行专用的图形指令。这些专用的图形指令涉及对视频 RAM 132 的位映射部分中的数据进行处理。专用的图形硬件 210 在中央处理单元 200 的控制下工作,以使得对于视频 RAM 132 中的数据能进行特定的有利的数据处理。

存储器接口 250 被耦连到总线 205,并进一步耦连到视频存储器总线 122。存储器接口 250 用于控制在图形处理器 120 和存储器 130 之间进行的数据和指令的通信。存储器 130 包含将在视频显示器 170 上显示的位映射数据以及控制图形处理器 120 工作的必要数据和指令。这些功能包含存储器存取的定时控制、和对数据及存储器的多路传输控制。在较佳实施例中,视频存储器总线包含多路传输的地址和数据信息。存储器接口 250 使图形处理器 120 能在适当的时刻在视频存储器总线 122 上提供特定的输出,以对存储器 130 进行存取操作。

最后,图形处理器 120 包含输入/输出寄存器 260 和视频显示控制器 270。输入/输出寄存器 260 双向耦连到总线 205,以便能在这些寄存器中进行读写。输入/输出寄存器 260 最好设在中央处理单元 200 的原有存储空间中。输入/输出寄存器 260 包含用来指定视频显示控制器 270 的控制参数的数据。根据在输入/输出寄存器 260 中存储的数据,视频显示控制器 270 在视频控制总线 124 上产生信号,为数字变视频的变换器 160 提供所需控制。输入/输出寄存器 260 中的数据包含:用于指定每个水平行中像素数目的数据;用于指定水平同步和消隐间隔的数据;用于指定每帧水平行行数的数据;和用于指定垂直同步和消隐间隔的数据。输入/输出寄存器 260 也可包含用于指定帧的隔行扫描类型和用于指定其它视频控制功能的数据。最后,输入/输出寄存器 260 也用来存储其它特定种类的输入和输出参数,这些参数在下面将更详细叙述。

图形处理器 120 以两种不同寻址方式对存储器 130 进行操作。这

两种寻址方式为xy寻址和线性寻址(linear addressing)。因为图形处理器120 同时对位映射数据和传统的数据和指令进行操作, 所以对存储器130 的不同部分可以不同的寻址方式最方便地进行存取。不论所选用的特定的寻址方式是何种方式, 存储器接口250 总是对所存取的数据产生适当的物理地址。在线性寻址中, 一个区域的起始地址表现为单一的多位线地址形式。该区域的大小由中央处理单元200 中的状态寄存器中的数据来确定。在xy寻址中, 该起始地址表现为一对x 和y 的坐标值。该区域的大小等于像素的大小, 即用于描述特定像素的特定数据所需要的位数。

现在转到图3, 在对本发明实施例的功能作详细描述之前, 简略地讨论下典型的图形存储器系统的存储器结构是有必要的。虽然有多种存储器结构和系统可以使用, 但如图3 中所示的一种结构更加典型, 它使用8 个V R A M (视频随机存取存储器) 存储器130 排成阵列。每个V R A M 存储器或单元具有四个部分(或称为板) 0、1、2 和3。每板的结构是, 只使用一根数据引线把信息写入该板。在使用32位数据总线的系统中, 例如数据总线122, 将有8 个V R A M 存储器(图3 中显示了两个), 每个V R A M 存储器有四个数据引出线连接到输入数据总线。

于是, 对于32位数据总线, V R A M 存储器130 用它的四根数据引线分别连接到数据总线0、1、2、3 引出线。同样, 下一个V R A M 存储器的四根数据引线0、1、2、3 分别连接到数据总线的4、5、6、7 引出线。所有剩余的六个V R A M 照此连接, 直到最后的V R A M 的四个数据引线连接到总线122 的28、29、30、31 引出线。

存储器的排列, 将使图形显示器的像素信息跨板依次贮存在相同的行中。假定系统的每个像素有四位, 那末每个像素的这些位存贮在

各自的VRAM存储器中。在这种情况下，像素0在第一个VRAM中，而像素1在第二个VRAM中。像素2到7的存储状况图中没有显示。像素8的像素信息存储在第一个VRAM中，仍然在行0中但占据其第二列（列2）。像素信息这样安排的理由，在了解信息且怎样从存储器中检索出来的以后，就完全清楚了。

继续参阅图3，每个VRAM的板上一串行寄存器140，用于从存储器行中移出信息。这些寄存器的输出连接到总线145，其方式与数据输入引线连接到输入总线的方式相同。于是，存储器行中的数据，譬如说行0的数据，将移入寄存器140中，这发生在8个存储器阵列的每个板中。

当在某一时刻察看数据输出总线145时，每个移位寄存器中的第一位会出现在总线上。于是，假定行0正在输出到总线，那末存储器130的行0的位A0（板0）会出现在总线的引线1上。总线145的引线1有行0的位A0（板1），引线2有行0的位A0（板2），而引线3则有行0的位A0（板3）。这些位的后面是下一个VRAM的各个位。于是，在第一个时刻，数据总线145上有四个位形成像素0。随后的四位形成像素1，再随后的四位形成像素2。这样继续下去，直到在数据总线145连续的各个引线端上出现由32位形成的0—7的8个像素。把这些位提供给图形显示器，接着移位寄存器将全部移动一位以便给总线提供下一组8个像素的像素信息，即像素8到15。然后这种移位继续进行，直到整个行被移出。之后又选择新的一行送入输出串行寄存器。

迄今为止，我们假定每个像素的信息位是4位。如果像素信息变为譬如说是8位，那末对每个像素必须用两个VRAM。也会稍微改变。也请注意，存储器大小和结构可继续变化，而且所示的大小和结构仅用作举例说明，本发明能够使用众多的不同存储器结构和不同的

像素大小。

如前面所讨论过的，每个存储器的每块板上的串行寄存器有512位长，因此对于每一个从存储器到串行寄存器的读出周期有16384位可传输到显示器。这16384位假如每个像素包含8位，可代表2048像素的数据。然而，假定每根扫描线仅需要1280个像素。于是在存储器的每根线上，每行存储器有268个像素不能得到显示。这个存储器，随着我们这儿的深入讨论将更清楚，要作别的用途是困难的，而且在效率上是浪费的。

为解决这一问题，可将这些串行输出寄存器对半分开，并且每一半均用来输出VRAM的数据，我们知道存储器共使用了32个移位寄存器，但我们将集中讨论存储器的一个板，因为所有板的工作方式是相同的。寄存器的两半称作A半和B半。串行寄存器的目的，是从存储器取出一个完整的屏幕（存储器）的一行，并将该行一个像素接一个像素平滑地显示在屏幕上。

如上所讨论，如果使用单一的非对分式的串联寄存器，那末显示器一个完整扫描行的信息必须从存储器130移入串行寄存器，然后以屏幕时钟速率移到屏幕上。这样就要求存储器的每一个仅仅包含屏幕信息的一行（或其整数倍）。若使用对分式的串行寄存器情况就不是这样了，我们将看到在这种寄存器中，当A段的位位移到屏幕时，其它位可装入B段；而当B段的位位移到屏幕时，其它位可装入A段。

现有转到图4，它显示了一个图形屏幕401，在其表面的横向上有40个像素，向下有若干行。必须清楚：这儿所用的数字仅用作说明，与实际图形屏幕的表面（横向典型像素数为1280个毫无共同之处。实际的数目如此之高，以致如果引例中使用接近于通常系统中实际使用的数字的话，本发明的操作会变得十分累赘。这也适用于随后要进行的存储器501（见图5）的讨论，而且使用实际数目的系统安排只会使得讨

论含混不清。事实上，如所看到的那样，作为讨论用的存储器501，在像素方面比屏幕401 有较少的列容量(16)。但在实际中，典型的情况是恰恰相反。

暂时离开一下上述讨论。一个典型的具有每行1280个像素和1024行的系统，其画面以每秒60次的速率刷新，所以每隔12.7ns必须显示一个像素。使用8 位的像素，即用两个4 位的 V R A M 提供每个像素的数据，将有4 组 V R A M 连接到32位总线上。要求 V R A M S 的时钟每50.8ns 出现一次，即频率为19.6M H z 。对于这样高速的数据移动，任何小小的暂停(如对串行寄存器重新加载) 都很容易被察觉。

现在回到图5,图中存储器501 具有4 个板。假设每个像素有4 位。为了说明问题，这儿我们也假设仅使用两个这样的存储器单元，一个单元包含偶数像素，而另一个(未画出) 包含奇数像素。这导致使用8 位或8 条引出线的总线，每个存储器单元对应4 位。我们仍然假设这种存储器仅有16列、标号0 到15，于是0 行是标号A₀ 到A₁₅，而1 行是标号B₀ 到B₁₅。如果我们进一步简化我们的讨论、把讨论限定在仅包含偶数像素的存储器，那么可认为A₀ 位代表0 像素的数据，而A₁位代表2 像素的数据。因此可推论在看不到的第二个 V R A M 中的A₀位将包含1 像素的信息。

遵照这一很不现实的但可作说明用的实施例，在A 行中将产生像素0—30(偶数) 的信息，在B 行中产生像素32—62(偶数) 的信息，如此等等。如图5 中所示那样。

现在让我们假设需要把代表屏幕第二行所必需的像素—第40—79 屏幕像素(图4) 的信息送到屏幕上。

为完成上述任务，该系统将把对存储器B 行寻址的指令位送给存储器，因为像素40—79的信息，如上面讨论那样，贮存在图5 的存储器B 行和C 行中。

上述操作的结果是把B行的像素32—62的像素信息装入串联寄存器中。这表明在图6中。然而，如果整个寄存器内容移位到屏幕上，则 B_0 位到 B_3 位也被位移，因为这些位是属于屏幕0行上的像素32—38（见图4），则结果产生了麻烦。为避免这一问题，控制存储器传送的处理器（图中未画出），将对移位的正确开始位置保持追踪，而且把这种追踪信息作为前述指令的一部分提供给存储器。这一位置称为分接点（the lap point）。

为了控制对分式寄存器的操作状况，必须知道应在何时对寄存器的第一部分重新装入数据，这个时间应该是：当数据正在从第二部分移走并且数据已经从第一部分移走时，或紧接在回扫间隔之后留在第一部分中的数据现在属于屏幕的先前一行时。当然也必须知道何时应对寄存器的第二部分装入数据，它应该是在数据已从第二部分读出之后，正从第一部分读出时。为了完成上述功能，用一个计数器对在给定时刻上工作的串行寄存器的位置保持追踪。为了计数器的正确运行，必须知道寄存器中移位的第一个数据的起始点（分接点）。这是很必要的，因为如上所讨论的那样，该起始点未必是存储器行的起始点。必须采用若干步骤逐行地对计数器进行校准以控制串行寄存器的两个半部的装入和重装入。

串行寄存器的控制是这样的：当寄存器的第一半完成数据发送时，它即被清除并且重新装入，所以当数据位从寄存器的第二半被发送时，新的数据位便能装入第一半寄存器。实际上如果首先要发送的位是处在第二半寄存器即B半中，那末A半将必须立即重新装载。这一情况是必须确定的。这种确定是由提供给存储器的寻址信息完成的，它取决于指定一个地址所必需的位的位置和位的数目。

作为上述情况的一个例子，让我们看看图9和图10中的某些典型的寻址位的结构。图9表示在3个存储体选体位和5个混合（miscel

laneous) 寻址位之前有10位行地址和10位列地址。图10表示在8 位行地址和8 位列地址之后仅有5 个混合寻址位。

掩蔽码由用户建立以适应系统的寻址结构。图11表示了适用于图9 寻址结构的一个掩蔽码，而图12表示适用于图10结构的一个掩蔽码。图13表示适用于我们讨论的系统的掩蔽码，即在两个选体位前有3 个分接点位(16 个可能的列，每半个移位寄存器对应8 列)。加上这些选体位且为了便于讨论。

图14展示了怎样使用这些掩蔽码，图15到20是我们的举例说明。

图15表示行与列的寻址位分别为存储器的1 行4 列，应该记得，它就是所选的屏幕行中第一个像素40所在的地址。图15中所描述的位字也包含其它寻址位0 --4 和存储体选择位5 --6 。分接点位装入分接点寄存器91。分接点可以定义为在寄存器中第一个读到总线上去的那个位的位置。该分接点可从图15的寻址信息中计算出来。在我们的举例中，第一个5 位地址(0—4)不予考虑，因为就设计而言它们对所有寻址结构都是一样的。接着的十三位地址被传送给图16的分接点寄存器91。

如图17和18所示，象图14所控制的那样，由我们的举例系统(图13) 产生的掩蔽码93被复制到掩蔽码移位寄存器92中。该掩蔽码用于在选体码的位数变化时调整分接点的位置。在我们的举例中，有两个选体位，于是该掩蔽码的前两位为0 。然后时钟将寄存器92和91向右进行移位，直到移位寄存器92(图19) 在最右面的位上出现1 为止。这一操作用来把存储选体位从分接点中移掉，于是分接点变成100, 如图20从寄存器91中看到的那样。

然后这100 装入分接点计算器94(图21) 。由移过位的掩蔽码92(图19) 确定在计数器94中哪些位是有意义的。这一分接点规定了串行寄存器中第一个应该读到数据总线的位置，从图6 中看到，该分接

点相应于由A半寄存器中的 B_4 位所控制的像素40。因为图15中列地址最左边的位为0,所以选择A寄存器而不选B寄存器。当列地址的最左面位置为1时,则选择串行寄存器的B半。

一旦移位后的分接点被选中,则时钟2001与存储器移位时钟一起工作,用来增加使分接点移位寄存器增值并从串行寄存器读出数据。于是,当分接点寄存器包含全部111位时,则意味着图6 A半寄存器的111位置的数据将被读到总线上。这对应于像素46,存储器位 B_7 。当移位B半寄存器开始时,则分接点计算器溢出到000,此时存储器位置 B_8 到 B_{15} 将依次送到图形显示器,请注意刚才描述的寄存器操作并不控制实际数据的移出,而是控制把数据重新装入串行寄存器。

此时,如图7所示,A半寄存器被清另且存储器位置 C_0 到 C_7 ,即存储器的下一行的信息装入A半寄存器。这种交替操作继续进行直到屏幕到达该行的末端、即像素79送到屏幕时为止。这种半行重新装载需要一个地址,该地址指向要重新装载的半行的第一位。这种地址来自“可增量复制的行地址”95。当寄存器91装入寄存器90的内容时,寄存器95也从寄存器90装入。接着在列地址的最左位增加增量1以指向下一个半行。寄存器93用来确定增量位的位置(在最左一个1的左面)。当该地址输出时,寄存器93也用来保证使该点以右的所有位为0(意味着分接头地址为零,指向移位寄存器中的第一位)。每次计数器溢出时,这一寄存器中的地址开始输出,之后再开始增量。

于是,当分接点时钟再次达到111时,并当像素62、即存储器位置 B_{15} 低于像素79时,分接点计数器复位到000,并且如图8所示,作为存储器的 C_0 到 C_7 从A半寄存器输送到总线。此时,B半寄存器装入存储器位 C_8 到 C_{15} 。而当时钟再次达到111时,屏幕也达到了回扫期,寄存器便复位以便由处理器确定把下一个满行读到屏幕上。此时,上述过程将重复进行并计算出新的分接点。

如果新的分接点表明要读出的第一位处在 B 半寄存器中(如果下一行是像素行 80 到 119 就会出现上述情况), 那末 A 半寄存器如图 8 所示将在位置 C_8 处显现分接点。这意味着 A 半寄存器必须立即清零且装入存储器位 D_0 到 D_7 , 以准备好让分接点计数器再次达到 111 并且翻转, 借以追从从第一半即 A 半寄存器中读出的数据。

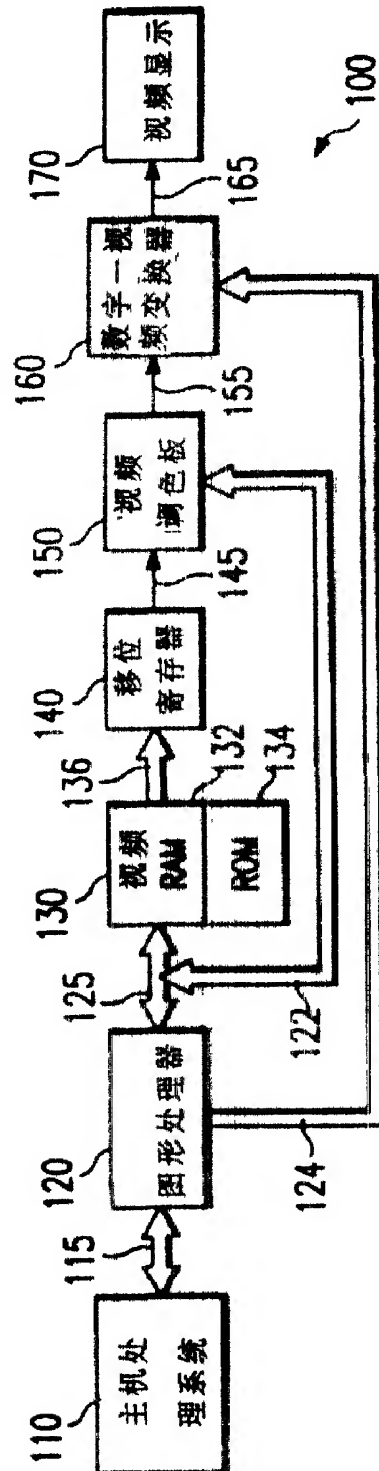


图1

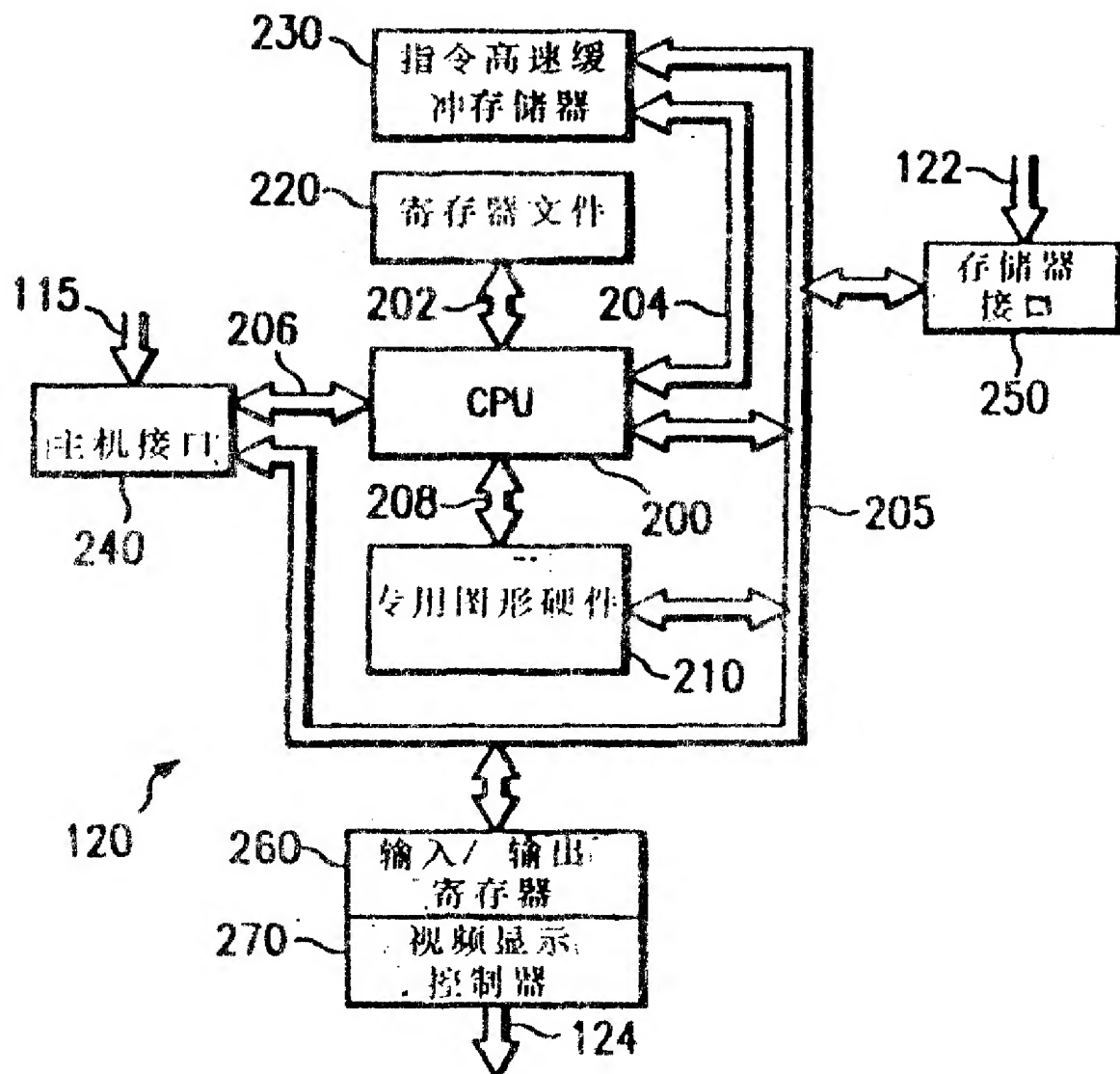


图 2

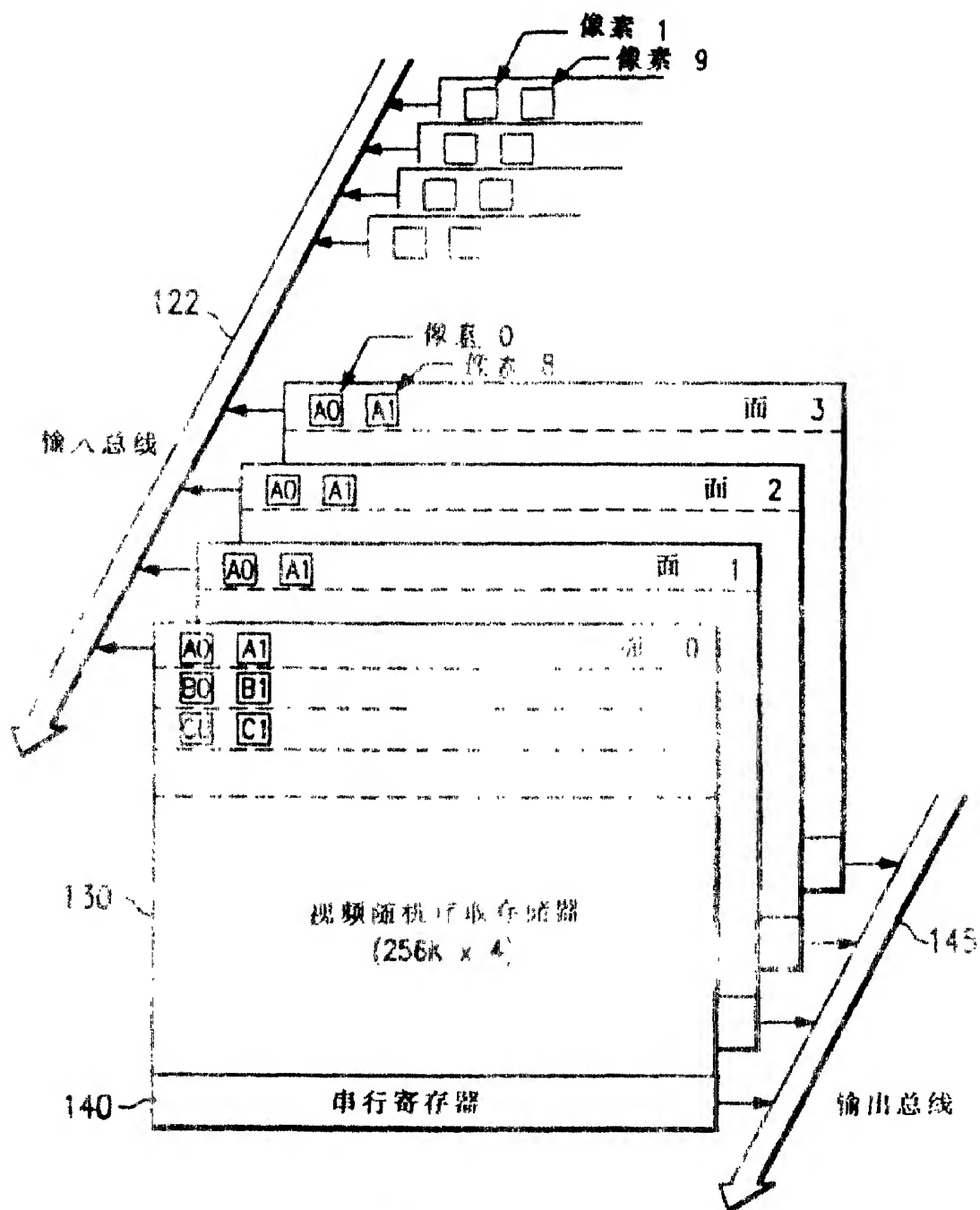


图 3

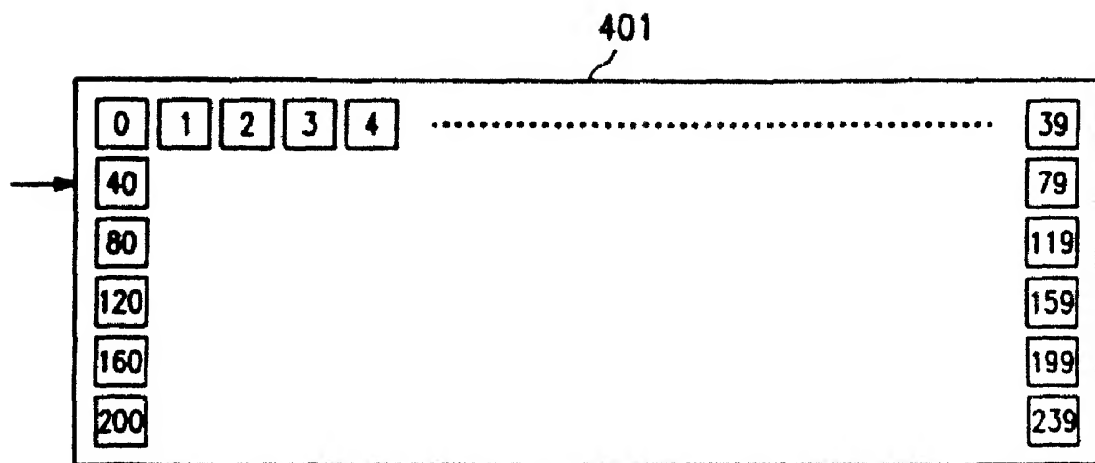


图 4

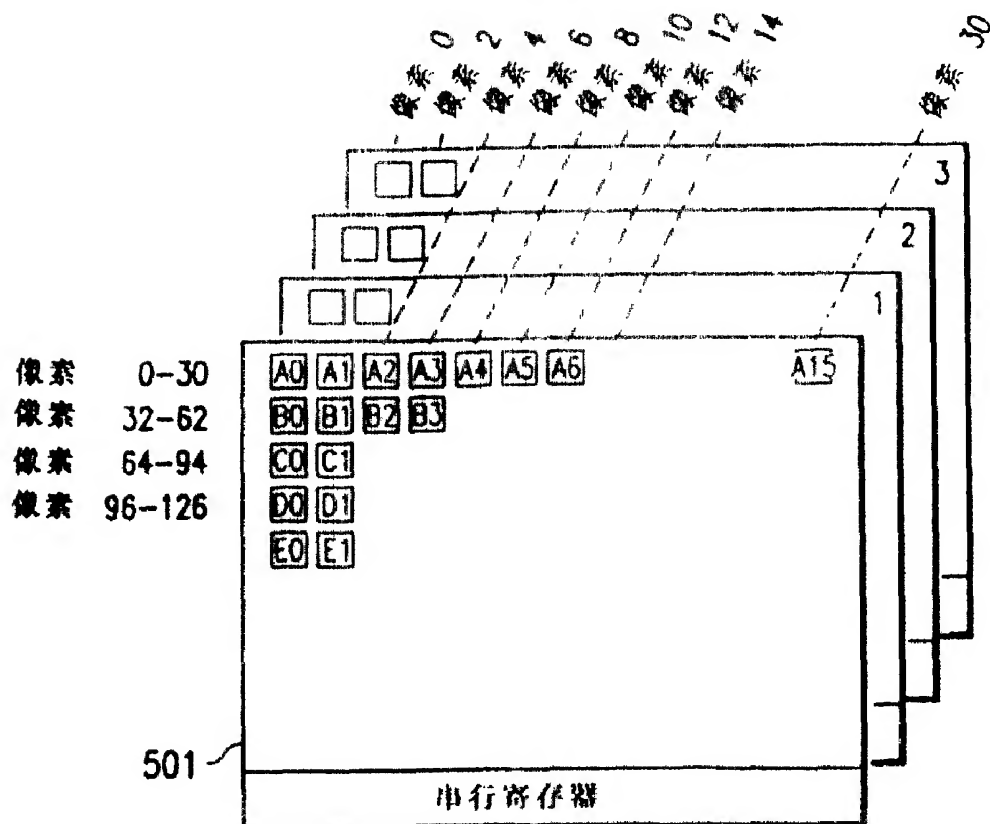


图 5

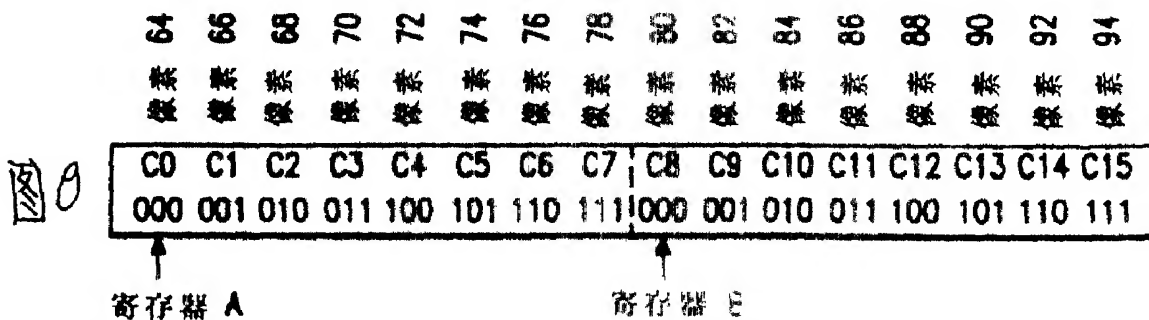
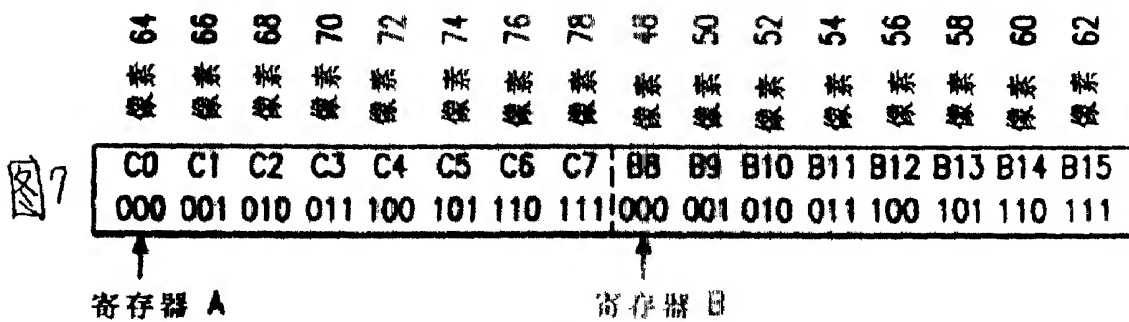
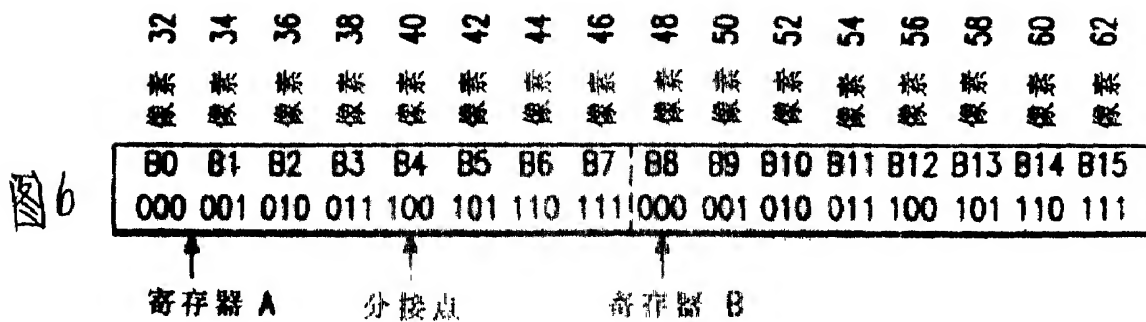
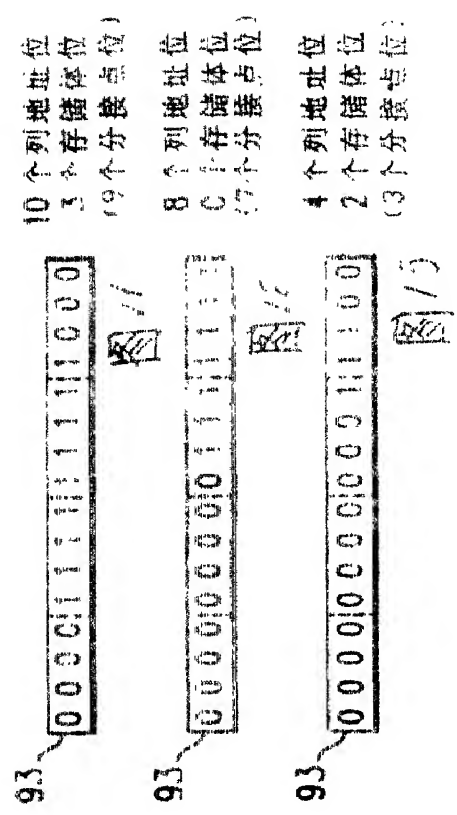




图 9



图 10



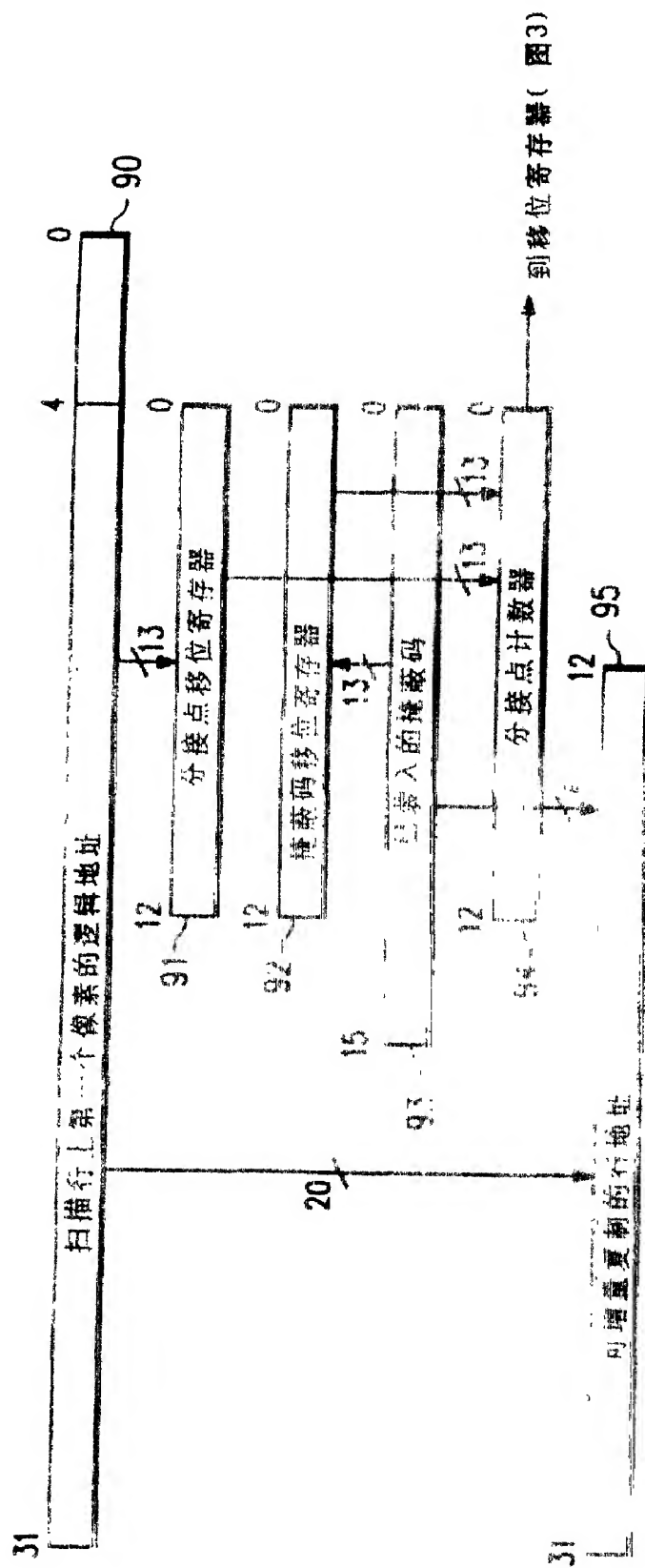


图 14

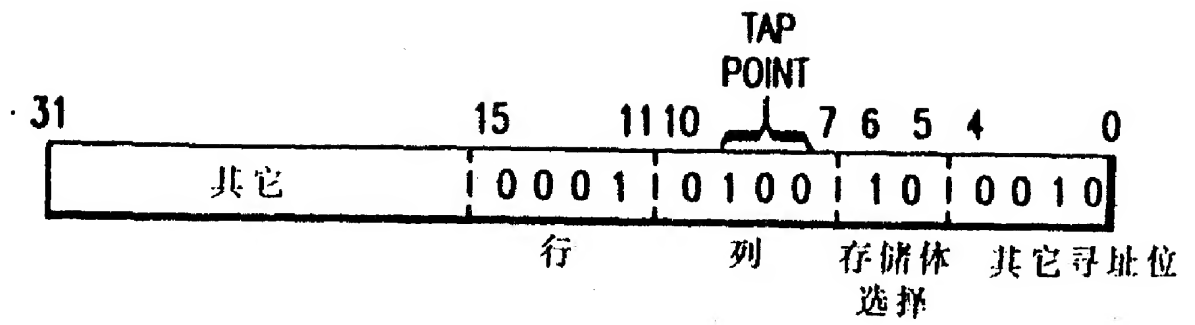


图 15

